EXHIBIT A

Java Card™ 2.1 Virtual Machine Specification

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Final Revision 1.1, June 7, 1999

Preface

Java Card™ technology combines a subset of the Java programming language with a runtime environment optimized for smart cards and similar small-memory embedded devices. The goal of Java Card technology is to bring many of the benefits of Java software programming to the resource-constrained world of devices such as smart cards.

The Java Card platform is defined by three specifications: this *Java Card™ 2.1 Virtual Machine Specification*, the *Java Card™ 2.1 Application Programming Interface*, and the *Java Card™ 2.1 Runtime Environment (JCRE) Specification.*

This specification describes the required behavior of the Java Card 2.1 Virtual Machine (VM) that developers should adhere to when creating an *implementation*. An implementation within the context of this document refers to a licensee's implementation of the Java Card Virtual Machine (VM), Application Programming Interface (API), Converter, or other component, based on the Java Card technology specifications. A Reference Implementation is an implementation produced by Sun Microsystems, Inc. Application software written for the Java Card platform is referred to as a Java Card applet.

Who Should Use This Specification?

This document is for licensees of the Java Card technology to assist them in creating an implementation, developing a specification to extend the Java Card technology specifications, or in creating an extension to the Java Card Runtime Environment (JCRE). This document is also intended for Java Card applet developers who want a more detailed understanding of the Java Card technology specifications.

1.3 Java Language Security

One of the fundamental features of the Java virtual machine is the strong security provided in part by the `class` file verifier. Many devices that implement the Java Card platform may be too small to support verification of `CAP` files on the device itself. This consideration led to a design that enables verification on a device but does not rely on it. The data in a `CAP` file that is needed only for verification is packaged separately from the data needed for the actual execution of its applet. This allows for flexibility in how security is managed in an implementation.

There are several options for providing language-level security on a Java Card technology enabled device. The conceptually simplest is to verify the contents of a `CAP` file on the device as it is downloaded or after it is downloaded. This option might only be feasible in the largest of devices. However, some subset of verification might be possible even on smaller devices. Other options rely on some combination of one or more of: physical security of the installation terminal, a cryptographically enforced chain of trust from the source of the `CAP` file, and pre-download verification of the contents of a `CAP` file.

The Java Card platform standards say as little as possible about `CAP` file installation and security policies. Since smart cards must serve as secure processors in many different systems with different security requirements, it is necessary to allow a great deal of flexibility to meet the needs of smart card issuers and users.

1.4 Java Card Runtime Environment Security

The standard runtime environment for the Java Card platform is the Java Card Runtime Environment (JCRE). The JCRE consists of an implementation of the Java Card virtual machine along with the Java Card API classes. While the Java Card virtual machine has responsibility for ensuring Java language-level security, the JCRE imposes additional runtime security requirements on devices that implement the JCRE, which results in a need for additional features on the Java Card virtual machine. Throughout this document, these additional features are designated as JCRE-specific.